

in your application. Click [here](#) to obtain the valid key.



# GIS

## Succinctly

by Peter Shaw

Created with a trial version of Syncfusion PDF library.



Technology Resource Portal

# GIS Succinctly

---

By  
**Peter Shaw**

Foreword by Daniel Jebaraj

Created with a trial version of Syncfusion PDF library.



# Chapter 1 So, what exactly is a GIS?

To most people, what they see as a GIS is in fact just the front-end output layer, such as the maps produced in Google Maps, or the screen on a TomTom navigation device. The reality of it all extends far beyond that; the output layer is very often the end result of many interconnecting programs along with massive amounts of data.

A typical GIS will include desktop applications used to visualize, edit, and manage the data, several different types of backend databases to store the data, and in many cases a huge amount of custom written software tools. In fact, GIS is one of the top industries where a programmer can expect to write a very large amount of custom tooling not available from other companies.

We'll explore some of the applications in detail soon, but for now we'll continue with the 100-foot view. A typical GIS processing setup will look something like the following:

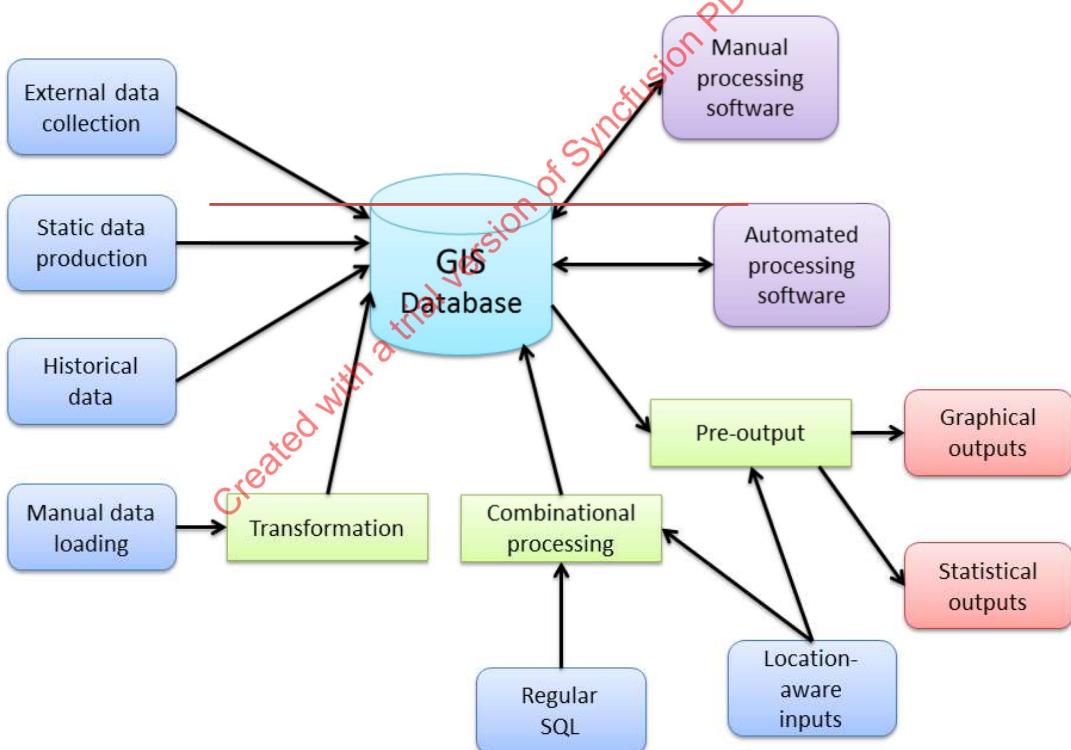


Figure 1: Typical GIS processing setup

As you can see in the diagram, the central part is very often the database itself with a huge number of inputs and processing steps. Finally, the output layers (shown in red) are what people usually associate with being a GIS.

Based on this, we can see that the database is the center of the universe when it comes to GIS.

# A Breakdown of the Components

Looking at the diagram in Figure 1, we can see that there are a number of parts that have specific meanings. We have our inputs (blue), outputs (red), in-place processing (green), and end processing (purple). At this point you might be asking yourself, "How is this different from any other data-centric system I deal with?" and you'd be right to do so. The main difference here is that in a typical GIS, you have to design everything in each component from the very beginning. With a regular data-centric system, many of the components are often optional or are combined into multifunctional components.

For a typical GIS, none of what you see in Figure 1 is optional, except for possibly your inputs. Even then, the components you'll most likely see omitted are manual and historical data.

So what do these separate entities entail, and why are they often not optional?

## External Data Collection

As the name suggests, this is the process of gathering external data specific to the system being designed. Typically this will come from custom devices running custom software (often embedded or small scale) designed to create input data in a very specific form for the system it is being used in. The lack of any in-place processing generally means the data produced is in a format that is already acceptable in the setup.

This component is typically satisfied by many diverse pieces of technology, and in most cases requires some training to use correctly. You'll often see things like digital surveying equipment or specialized GPS devices fitted to vehicles, which in many cases will often feed data back in real time using some kind of radio connection.

## Static Data Production

Like external data, this process normally gathers data in a specific format for the system it is being used in. Unlike external data however, you will generally find that static data is produced in-house by scanning existing paper maps or digitizing features from existing building plans, for instance.

Like external inputs, static data is often produced using custom software and processes specific to the business.

## Historical Data

Because of the size and amount of data produced in a typical GIS setup, there is often a need to back up data into a separate archival system while still maintaining the ability to work with it if needed. Often, data of this nature is created by planning authorities showing things like land use over time or recording where specific points of interest are. This is treated as a separate input because the data is usually read only, and similarly to external and static data, was at one time produced specifically for the system.

## Manual Data Loading

While the name of this type of input may suggest the same as external data, the actual data obtained in this step is usually very different. Data coming into the system via this input will often be in the form of pre-provided data from a GIS data provider. In the United Kingdom, this will often mean data provided by companies like Ordnance Survey. In the United States, this might mean data provided by institutions such as the U.S. Geological Survey or TIGER data from the U.S. Census Bureau.

At this step, wherever data is obtained from, it's almost guaranteed that it will need to be transformed into a format that is useable in the GIS it's destined for. More often than not, it will need to go through some kind of in-place process before it's useable in any way.

## Regular SQL Queries

Since most GIS have a large database at the center of them, SQL still plays an important role and probably always will. However, in GIS terms, these queries not only involve the normal SQL that you're used to seeing in a database management system, but also geospatial SQL. We'll cover GIS-specific SQL a little later on; for now, inputs here are usually generated from things like search queries.

As an example, when you type the name of a place or a ZIP code into Google, Bing, or Yahoo Maps, the web application you're looking at will most likely turn your search into a query that uses geospatial SQL to examine data in the core database. This, in turn, will be combined with other processes to produce an output, which in this case will usually be a map displaying the location you searched for. Another example might be an operator in an emergency services control room entering the location of an incident, and combining that with the known locations of nearby emergency vehicles to aid in making a decision as to which vehicle to send to the incident.

## Location-Aware Inputs

The last input type is probably the one that is familiar to most people. Location-aware data most often comes from the GPS input on a mobile phone or other GPS-enabled device. It is generally common latitude and longitude information. We'll cover this more when discussing NMEA data.

## Graphical Outputs

Now we move to the output layers, the first of which is the graphical one and what most people are familiar with. Output data here is very often in the form of a raster-based map with all operations performed to produce a single output tile in the form of a standard bitmap (such as a .jpeg). However, far more is involved than simple map tiles. Graphical outputs can, and very often are, produced in various vector formats, or as things like AutoCAD drawings for loading into a CAD or modeling package. In fact, even in web environments where people are used to seeing bitmap tiles, it's common for graphical output to take the

form of SVG or KML data combined with a custom Google Maps object. Raster tiles are just the tip of the iceberg.

## Statistical Outputs

Outputs in this group are the complete opposite of graphical outputs. Data is often the by-product of several GIS–SQL operations based on the input data and processes going on within the system. Just like general database data, from this output you'll get facts and figures that can be used to report statistics to management or marketing teams. The reason we treat this separately, however, is because of the nature of the information.

While you might be tempted to just say, "It's only numbers," in some cases it's numbers that have no meaning unless there is some GIS input involved. As an example, let's say we have a number of geographic areas representing plots of land, and with each of those areas we have a monetary value for that plot.

We can easily say, "Give me the values of each plot in descending order," enabling you to see which is the most expensive piece of land overall. This is where the difference stops, however. Let's say we now know that all land in a district has a 1% tax for every square meter a plot consumes. We know by looking at a graphical output of the map that the visually bigger areas are going to be more expensive, but you can't convey that to a computer.

You can, however, ask using GIS–SQL for a statistical analysis based on a percentage of the land's plot value multiplied by however many square meters are in the defined area boundary.

## Manual Processing Software

Anything in the system that requires an operator and some software to make changes falls under the category of manual processing software. Typically, this is both an input and an output because in most cases this involves changes being made to the underlying data manually.

This is usually the area where you'll see large GIS packages such as ESRI, DigitalGlobe, and MapInfo used. We'll cover some of these later. An example of what might be performed at this stage is boundary editing. Let's say that you added some town boundaries as area definitions several years ago, and since they were first added the towns have increased in size. You would then find a GIS expert who, with his or her chosen software and some satellite imagery, would edit your boundary data so that its definition better fits the newly expanded imagery.

## Automatic Processing Software

Operations running at this stage are generally not much different than those being run manually. The reason we see a clear separation is because some processes simply cannot be automated and need a human eye to pick out details. Going back to our previous

example of the town boundaries, it's not beyond imagination that a process can be defined to analyze an aerial image and determine if boundaries need to be removed.

Most often, however, automatic editing is used to perform tasks such as drift correction or height and contour changes due to earth movement.

## Transformation Tasks

As mentioned in the discussion of manual data input, when obtaining data for incorporation into a GIS, the data will rarely be in a format suitable for inclusion in the system.

Making the data usable may involve something as simple as a coordinate transform, or something as complex as combining multiple datasets based on common attributes and more. Transformation processes can and often do seriously affect the overall data quality, and many systems can end up with a lot of deeply rooted problems caused by mistakes when transforming data.

In the U.K., these processes are almost always seen when working with latitude and longitude coordinates, as nearly all the data supplied by U.K. authorities will be in meters from the origin, rather than degrees around the center.

## Combinational Processing

Combinational processing is generally in-place processing that is the result of various input operations. It's not too different from using a join in a regular database operation. The result is a combination of processes and input data steps that ultimately work in real time to produce a defined input data set.

## Pre-Output

Last but not least is the pre-output step. As the name suggests, this is the final processing required before the output is useable. A pre-output process may include transforming an internal coordinate system to a more global one; for example, U.K. meters back to a global scale, or converting a batch of statistics to a different range of values. Location-aware inputs are often included in this step, typically in a navigation system. For example, a location's graphical representation could be combined with current mapping to produce a visual output for a tracking map.

## The Database

So just what makes a GIS database so different from a normal database? Honestly, not much. A GIS database is simply specialized for a particular task.

A better way to illustrate what makes a GIS database unique is to look at the growing world of big data. These days, it's hard not to notice how much noise is being made by NoSQL and document-centric database providers. These new-breed databases fundamentally do the

same things as a normal database, but use specialized processes that perform particular operations in better, more efficient ways.

Looking at a GIS database through the lens of a non-GIS connection, the geometric data is nothing more than a custom binary field, or blob, that the software and processes working with the system know how to interpret. In fact, it's possible to take a normal database engine and write your own routines, either in the database or in external code, to perform all of the usual operations you would expect but with GIS data.

In general, when a database is spatially enabled, it will have much more than just the ability to understand the binary data added to it. There will be extensions to the SQL language for performing specialized GIS data operations, new types of indexes to help accelerate lookups, and various new tables used to manage metadata pertaining to the various types of GIS data you may need to store.

I'm not going to list every available operation in this book, only the most important things you need to know to get started. At last count, however, there are more than 300 different functions in the last published OGC standards.

## OGC What?

The OGC standards are the recommendations set by the Open Geospatial Consortium. They define a common API, a minimum set of GIS-SQL extensions, and other related objects that any GIS-enabled database must implement to be classified as OGC compliant. Because of the diversity of GIS and their data, these standards are rigorously enforced. This enables nearly every bit of GIS-enabled software on the planet to talk to any GIS-enabled database and vice versa using a common language.

Note that when selecting a database to use, there are many that claim to be spatially aware but are not OGC compliant. Prime examples are MS SQL and MySQL.

In general, MS SQL features the OGC-ratified minimum GIS-SQL and functional implementation, but its calling pattern varies significantly from most GIS software. MS SQL also features changes to column names in some of the metadata tables, which means most standard GIS software cannot talk to a MS SQL server. Note also that MS SQL didn't add any kind of GIS extensibility until 2008, and even in the newer 2008 R2 and 2012 versions, the GIS side of things is still not completely OGC compliant.

MySQL has similar restrictions, but also treats a number of core data types very differently, often leading to rounding errors and other anomalies when performing coordinate conversions. You can find the full list of OGC standards documents on the OCG website at <http://www.opengeospatial.org/standards/is>.

A good place to look for information comparing various databases is on the BostonGIS website at [http://www.bostongis.com/?content\\_name=sqlserver2008r2\\_oracle11gr2\\_postgis15\\_compar\\_e#221](http://www.bostongis.com/?content_name=sqlserver2008r2_oracle11gr2_postgis15_compar_e#221).

There are also a number of other good starter articles on the site. The downside is that the site is cluttered and sometimes very hard to read.